

United States Application

Entitled: SOFTWARE TOOL FOR CREATING AN INTERACTIVE LIST

Inventor: Thomas M. Sirhall

Software Tool for Creating an Interactive List

Field of the Invention

- The present invention relates to a web-based learning environment. More particularly, the present invention relates to an interactive software tool for testing a user from a remote location.

Background of the Invention

- The Internet has revolutionized educational opportunities by accommodating on-line educational courses. On-line educational courses can provide a diverse educational curriculum without imposing long-distance travel on students. Furthermore, on-line educational courses accommodate students with a wide variety of lifestyles, provide learning at an appropriate pace for a student and can accommodate a student's time constraints. A student may enroll in a periodic on-line educational course to learn about a new technology or other subject of interest without disrupting participation by the student in other activities, such as a full-time jobs or activities with their family.

- However, a drawback of current learning opportunities over the Internet is the difficulty in testing a student. In the traditional classroom setting, testing has proven to be extremely beneficial to the learning process of a student. Testing helps assess a student's retention and absorption of presented information and allows an instructor to chart a student's progress through the learning process. Testing also serves to reinforce and emphasize the presented information in the mind of the student. Testing further provides feedback to an instructor or course creator regarding the clarity and comprehensiveness of the material presented. Without an effective testing process, a distance-learning student may be merely a passive viewer of information, rather than an active participant in the educational process.

Summary of the Invention

- The present invention provides an interactive software application for testing a user's knowledge in a web-based learning environment. According to an illustrative embodiment, the present invention provides an interactive software tool embedded in a course page of a web-based curriculum. The software tool provides a question to a user

regarding the information presented in the course and instructs the user to provide an answer to the question. The software tool provides feedback to the user, and allows the user to revise an incorrect answer.

- 5 According to one embodiment of the present invention, a method is provided for use in an electronic device is that provides an on-line educational course, including the steps of providing an interactive list software tool, wherein the software tool generates a question and an answer box to a user to select at least one answer to the question in the answer box and forwarding the list software tool from the electronic device to a remote
10 client.

- According to another embodiment of the present invention, a method is provided for use in an electronic device is that provides an on-line educational course, including the steps of receiving a request for a Web page at the electronic device from a remote
15 client and in response to the receiving step, sending a Web page containing a question and a list software tool embedded therein to the remote client, wherein the list software tool generates a graphical user interface (GUI) including instructions to a user to enter an answer to the question provided by the Web page.

- 20 A further embodiment of the invention involves a computer-readable medium for use in an electronic device that provides an on-line educational course, including instructions for running a list software tool for displaying a question and an answer box to a user, wherein the user can select at least one answer in the answer box to the question.

- 25 Another embodiment of the invention provides an electronic device for providing an on-line educational course including a processor, a display screen and memory including a Web page having an interactive list software tool embedded therein. The processor executes the list software tool to generate a graphical user interface on the
30 display screen, the graphical user interface displaying a question, and an answer box to a user to select at least one answer from the answer box to the question.

Brief Description of the Drawings

The foregoing and other objects, features and advantages of the invention will be apparent from the following description and the accompanying drawings, in which like reference characters refer to the same parts throughout the different views. The

5 drawings illustrate principles of the invention and, although not to scale, show relative dimensions.

Figure 1 is a block diagram of a distributed environment suitable for implementing an on-line educational course and running the list software tool of an illustrative embodiment of the present invention.

10 Figure 2 is a block diagram of a client machine suitable for use in the illustrative embodiment of the present invention.

Figure 3 illustrates a web page including the list software tool of the illustrative embodiment.

Figure 4 illustrates the list software tool when a user enters a correct answer to the question asked by the software tool.

Figure 5 illustrates the list software tool after the user has incorrectly answered the question.

Figure 6 illustrates the list software tool after the user has supplied a partially correct and partially incorrect answer.

20 Figure 7 illustrates the list software tool after the user has made the maximum incorrect attempts allowed to enter a correct answer.

Figure 8 illustrates the list software tool after the user has entered an incomplete answer.

Figure 9 is a flow chart illustrating the steps involved in creating a list software tool according to an illustrative embodiment of the present invention.

Detailed Description of a Preferred Embodiment

Figures 1 through 9, wherein like parts are designated by like reference numerals throughout, illustrate an example embodiment of a software application suitable for interacting with a user to test the user on information presented in an on-line educational course. Although the present invention will be described with reference to an illustrative embodiment shown in the figures, those skilled in the art will appreciate that the present invention may be implemented in a number of different applications and

embodiments and is not specifically limited in its application to the particular embodiment depicted herein.

The software application of the illustrative embodiment provides a mechanism by which a provider of an on-line network learning center can test a student on information presented in an on-line education course setting of an on-line network learning center environment. An “on-line network learning center” is an organization that provides instruction over a wired and/or wireless communication network, such as the Internet, an intranet, an extranet, a local area network (LAN) and a wide area network (WAN) and other networks enabling communication among computers. An on-line network learning center maintains a server having a domain name associated with the learning center. Administrators of the learning center develop, create, edit and/or administer an on-line educational course. An “on-line educational course” is curriculum, such as a training course for a new technology, that is delivered to a student over a wired and/or wireless communication network as described above. Course material in the on-line educational course may consist of many units or chapters and may include text, graphics, sound, movies and exercises to be performed by the student. Through the use of the software application, a list software tool, the student is presented with a question related to the information presented in an on-line course, followed by an answer box having correct and incorrect answer selections, allowing the student to select correct answer selections. The software tool provides feedback to the student informing the student if he has correctly answered the question. If the student is incorrect, the software tool invites the student to try again. After a predetermined number of attempts, the software tool automatically provides a correct answer to the student.

25

As used herein the term “question” refers to statements phrased as questions and also to statements not phrased as questions. In either case, the statement will preferably be phrased so that at least one particular response to the statement can be determined to be a correct response. The term “answer” as used herein refers to responses to the statement, regardless of whether the statement is phrased as a question.

The present invention can be implemented in an on-line educational learning center. The learning center may provide on-line opportunities for students to enroll in

educational courses pertaining to any number of different subject areas and to be tested on the information presented in the course. The exam questions that exist within the on-line learning center can be provided by the creator and host of the learning center, or can be provided by third parties (e.g. universities or vendors). A third party wishing to submit and create exam questions utilizing the interactive software tool of the present invention can access a learning center website, using a browser such as the HotJava browser from Sun Microsystems, Inc. of Palo Alto, California or Netscape Navigator from Netscape Communications. The browser utilized, for the purposes of the embodiments illustrated herein, supports the JAVA language, so that the third party can provide exam questions to the on-line learning center. JAVA is a trademark and registered trademark of Sun Microsystems, Inc. in the United States and other countries. A JAVA software tool may guide the third party through the process of creating exam questions for an on-line course within the on-line learning center structure.

JAVA is an object-oriented, platform-independent computer programming language and environment suitable for writing programs that run over the Internet. As used herein, the term "software tool" refers to a computer program. Optionally, the software tool may be an applet and designed to be executed from within another program. A software tool may be a small, specialized application written in the JAVA programming language that can be included in an HTML or XML page, much in the same way an image is included and can be executed in a Web browser. Software tools allow developers to add "interactive" content to Web documents (such as animation, page adornments, games, etc). Software tools can be downloaded from a Web server and executed within a JAVA-compatible browser (e.g. HotJava) by copying code from the Web server to a client. JAVA source code files (i.e. files with a .java extension) are compiled by a JAVA compiler to produce instructions into a format called bytecode (i.e. files with a .class extension), which can then be executed by a JAVA virtual machine (VM). JAVA VM's are available for different platforms and thus, help to provide "platform independence" for JAVA programs.

To run a JAVA software tool from a Web site, the site developer inserts software tool tags (i.e. <software tool> into an HTML document, an XML document or the code of some type of page description language, which instruct the browser to download the

appropriate classes from the Web server and then interpret the classes. "Classes" define JAVA program building blocks, called objects, as well as the behavior and attributes of the objects. Specifically, the tag identifies the JAVA classes needed for running the software tool and may also set parameter values for the software tool.

5

The illustrative embodiment of the present invention is described relative to a software tool in the JAVA programming language. However, the present invention is not limited to the specific embodiment. It should be noted that other languages can have different forms of a program that perform the same function, and the present invention can be used in a programming environment other than JAVA. Furthermore, the present invention is not specifically limited to a JAVA software tool executed from a Web browser, and may be implemented as various types of code modules in accordance with the teachings of the present invention.

15 Figure 1 illustrates a block diagram of a network 100 suitable for implementing an on-line educational course and running the list software tool of the illustrative embodiment of the present invention. The network 100 includes a Web server 101 and a plurality of client machines 102 connected to the web server 101 and running a JAVA-enabled Web browser 103. The Web server 101 and the client machines 102 are
20 connected to a network 100 via communications links 104. The Web server 101 includes a database 105, servlets 106, and Web pages 107 generated by the servlets 104. The Web browsers 103 on the client machines 102 locate and display Web pages. A Web browser further includes an HTML interpreter 108.

25 Figure 2 is a more detailed block diagram of an exemplary client machine 102. The client machine includes a central processing unit 109, a keyboard 110 enabling a user to enter data into the client machine 102, a video display screen 111 and a mouse 112 or other pointing device for controlling the movement of a cursor or pointer on the display screen 111. The client machine 102 further optionally comprises a network
30 interface 113 for connecting the client machine to the Internet, a modem 114 for formatting data to be transmitted over telephone lines, coaxial cable, or other communication lines, a decoder 115 for translating encoded data to its original format and client memory 116. The memory 116 contains Web pages 107 including embedded

software tools 117, a JAVA-enabled Web browser 103, and a JAVA class library containing the JAVA classes needed to run the software tools.

The client machine 102 can be a personal computer, a workstation, an Internet
5 appliance, a personal digital assistant, a cell phone, a set-top box with attached television, an intelligent pager or a wide variety of other items capable of communicating with a user.

To run the interactive list software tool of the illustrative embodiment of the
10 present invention, the Web server 101 sends an HTML document with an embedded list software tool to a client machine 102 of the network 100. The HTML interpreter then interprets the HTML document. The Java VM in the Web browser executes the list software tool and displays a Graphical User Interface (GUI) on the video display screen
111 of the client machine 102. The GUI, which will be described in detail with
15 reference to Figures 3 through 8, displays a question to a user and includes a box where the user can select one or more answers to the question.

Figure 3 illustrates a graphical user interface (GUI) 10 related to the list software tool according to an illustrative embodiment of the present invention. The illustrated list
20 software tool tests a user regarding information presented in an on-line JAVA tutorial. In Figure 3, a browser 11 illustrates the interactive list software tool. The GUI 10 of the illustrative list software tool displays question box 12 including a question related to information taught in an on-line course and instructions to the user. The GUI further includes an answer box 13. Although the answer box 13 is shown as approximately a
25 square having lined borders, many variations of organization of answers are within the scope of the invention, including no borders and answers scattered randomly within the GUI or in a row and/or column format. The user selects one or more answers to the question from the selection listed in the answer box 13 by clicking on each answer of the selection. By clicking on an answer again, the answer is deselected. A label 15 directs
30 the user to the answer box 13. The GUI includes function buttons 19, 21 to facilitate interaction with the user. When a user clicks on a function button, the software tool responds according to particular instructions stored in a software tool file.

In the GUI illustrated in Figure 3, an HTML document provides an envelope for the list software tool (i.e. the software tool is embedded in the HTML document). The HTML document provides the static background for the web page, including the instructions and/or question to the user. For example, the HTML document optionally includes a logo 14 and the question box 12.

In Figure 4, the user completes selecting one or more answers and presses the “check” button 19 to assess his selections. If the user has correctly selected the answer(s), the software tool provides positive feedback in feedback site 20 to the user. According to the illustrative embodiment, the positive feedback in response to a correct match reads “Correct! You successfully identified the valid vi commands. Close this window to continue with the course.” According to the illustrative embodiment, the software tool is contained in a “pop-up” web page that is accessed at a particular point in the on-line course, such as after the user finishes a lesson or another software tool. After the user has completed the activity, the software tool directs the user back to the on-line course.

However, if the user enters an incorrect answer, the software tool provides a negative feedback in the feedback site 20 and prompts the user to try again, as illustrated in Figures 5 and 6. As shown in Figure 5, when no correct answers have been selected in the illustrative embodiment, the negative feedback is “You have missed the answers. Update you selections.... 0 are correct.” Figure 6 illustrates an example involving one correctly selected answer and one incorrect answer. The list software tool includes a “reset” button 21 to allow the user to clear his entry. After a predetermined number of failed attempts, the software tool automatically provides the correct answer to the user, as illustrated in Figure 7. According to the illustrative embodiment, the correct answer to the question “Identify the vi commands from the list below and click the check button” includes “l”, “^g”, and “R”. According to the illustrative embodiment, after three attempts, the software tool notifies the user “Your last response was incorrect. The correct response has been made for you. Please close this window to continue with the course.” The software tool also disables the function buttons 19, 21 to prevent the user from subsequent action. According to the illustrative embodiment, there may be more than one correct answer for a particular question.

According to one embodiment, shown in Figure 8, the software tool further provides feedback to the user regarding an incomplete answer. If the user fails to enter a selection and then presses the "check" button 19, the software tool responds "You have missed the answers. Update you selections.. 0 are correct." or other similar command in the feedback site 20.

According to another embodiment, the software tool randomizes the answer selections upon subsequent re-loading of the software tool.

According to the illustrative embodiment of the present invention, the list software tool is embedded in a page within the on-line course. The page is an HTML, XML, or other page description language document including a link to the software tool. An HTML file is a hypertext markup language file conventionally used in creating documents on the World Wide Web or Internet. HTML defines the layout and structure of the file, using various tags and attributes. As discussed, the HTML file used in conjunction with the list software tool of the present invention can include exam questions and reference the JAVA list software tool that presents a question to the student, and then checks the answers.

An example of an HTML fragment suitable for implementation in the illustrative case includes the following code:

```
<html>
25 <body bgcolor="#ffffff" topmargin="0" leftmargin="0" marginwidth="0"
    marginheight="0">
    <table align="center" cellpadding="5" cellspacing="0" border="0">
    <tr>
    <td>
30 <table cellpadding="0" cellspacing="0" border="0">
    <tr>
    <td></td>
    </tr>
    </table>
35 <div align="center">
    <table width="98%" border="0" cellspacing="0" cellpadding="1"
    bgcolor="#000000">
    <tr>
    <td>
40 <table border="0" cellpadding="0" cellspacing="1" width="100%">
    <tr>
    <td height="22">
```

P5709
(SMQ-058)

```

5      <table width="100%" border="0" cellspacing="0"
      cellpadding="5" bgcolor="#FFFFFF">
        <tr>
          <td width="17%" valign="top" bgcolor="#000000"
10      align="center"><b><font color="#FFFFCC">Question:</font></b></td>
          <td width="83%" bgcolor="#e4e4e4">Identify the vi
            commands from the list below and click the check button. </td>
          </tr>
        </table>
10      </td>
        </tr>
        <tr bgcolor="#FFFFFF">
          <td height="22">
            <table width="100%" border="0" cellspacing="0"
15      cellpadding="8">
              <tr><td>&nbsp;</td></tr>
              <tr>
                <td>
20      <div align="center">
                <software tool
                archive="../../../SESTBTInteractivity.jar" code="ListSoftware tool"
                width="450" height="225">
                  <param name="file" value="listsoftware
25      tool.def">
                  <param name="TBTServletBase"
                  value=".">
                  <param name="TBTDocumentBase"
                  value="">
30      </software tool>
                </div>
              </td>
            </tr>
          </table>
35      </td>
        </tr>
      </table>
40      </div>

```

This code defines the attributes of the HTML document wherein the list software tool is embedded. Tag `<td width="83%" bgcolor="#e4e4e4">` of the code precedes the question and instructions to the user displayed in question box 12 in Figure 3. The HTML code can be run with the help of a browser such as HotJava or Netscape Navigator. The questions can be viewed by any user running the HTML code, using a browser to look at the source code. Generally, a user can easily view the source code for a web page by positioning a mouse pointer on the web page and clicking on the right button of the mouse. Therefore, according to the present invention, the answers to the question are typically included in a separate file and are not generally available to the user.

The HTML code further includes a software tool tag. The software tool tag instructs the browser to load a software tool having the attributes defined by the tag.

The browser calls a list software tool of an illustrative embodiment of the present

- 5 invention with the following software tool tag HTML code:

```
10 <software tool codebase="<prop servletname>content:/java/"  
archive="SESTBTInteractivity.jar" code="ListSoftware tool" width="500"  
height="450"> <param name="file" value="listsoftware tool.def"> <param  
name="TBTServletBase" value="<prop servletname>"> <param  
name="TBTDocumentBase" value="<subst contentPageDocBase>"> </software  
tool>
```

- 15 The browser loads the software tool class files for the list software tool into a JAVA interpreter and calls a definition file defined by the software tool tag and corresponding to the list software tool. The tag tells the software tool viewer or browser to load the software tool whose compiled code is in the file "ListSoftware tool" and sets the initial size of the software tool to 500 pixels in length and 450 pixels in height. The
- 20 "codebase" parameter specifies the base URL of the software tool (the directory that contains the software tool's code). The "archive" parameter of the tag describes one or more archives containing classes and other resources for the list software tool. The .jar files referenced in the archive are compressed versions of JAVA files, suitable for transporting over a network, such as the Internet. The "param name" parameters specify
- 25 a software tool-specific attribute. The first param name invokes a definition file for the list software tool. The second param name parameter, TBTServletBase, specifies the server for the software tool and the parameter TBTDocumentBase specifies the directory containing the SESTBTInteractivity.jar file.

- 30 The JAVA definition file written for the list software tool of the present invention includes the answers and other parameters of the software tool, which are hidden from the user. The software tool tag directs the browser to the definition file "listsoftware tool.def" stored on the server. The definition file defines the appearance of the software tool and includes a number of attribute tags. For example, a definition file
- 35 for the list software tool of the illustrative embodiment the present invention may include the following code, which is illustrative of one embodiment of the invention and is not to be construed in a limiting sense:

P5709
(SMQ-058)

```
@bgcolor
white

5  @correct
   (

   @correct
   ^g

10 @answer
   pwd

   @answer
15 runvi

   @correct
   R

20 @numberofanswers
   5

   @numbercorrect
   3

25 @multipleselection
   on

   @correctfeedback
30 Correct! You successfully identified the valid vi commands. Close this
   window to continue with the course.

   @incorrectfeedback
35 You have missed the answers. Update your selections.
```

The definition file tags define particular parameters of the software tool. In the example set forth above, “bgcolor” defines the color to be displayed as a background. As defined by the illustrative definition file of the list software tool, the background color is set to white. The tag “correct” defines the text to be associated with a correct selection. The tag “answer” defines the text to be added to the answer box 13 as incorrect answer selections. As in the illustrative embodiment, more than one correct tag and more than one answer tag may be included. The “numberofanswers” tag defines the total number of answers to be available within the answer box 13, including both correct and incorrect answers. The “numbercorrect” tag defines the number of correct answers to be provided in the answer box 13. The “multipleselection” tag is set to open “on” if multiple answers are correct and is set to “off” if only a single answer is correct. The tag “correctfeedback” defines the feedback to be given to the user when the user

enters a correct answer. Finally, the tag “incorrectfeedback” defines the feedback to be given to the user when the user enters an incorrect answer.

- The definition file may further include a tag defining the maximum number of
- 5 attempts a student can make before the software tool automatically provides the correct answer. In general, an exam may include a several questions, and the maximum number of attempts defined by the software tool can vary according to the difficulty of each question. For example, an exam may include two questions, the first of which may be deemed by the administrator to be easier than the second. In such a case, the maximum
- 10 attempts may be set to a smaller number for the first question than the second question. This provides the student more opportunities to obtain the right answer for the second, more difficult question.

- One advantage to providing the answers to the exam question in a separate file,
- 15 instead of the HTML source file that references the JAVA software tool, is that a user cannot cheat by looking for the answers to a question in the source file, which is usually freely readable. Second, the JAVA software tool requires the correct answer, to check whether or not a student’s reply is accurate, but does not require the question. Thus, it is convenient to utilize a separate file as input to the JAVA software tool that contains
- 20 answers but no questions. The definition file can be altered, and subsequently used by the JAVA software tool, without having to recompile or rejar the software tool. Hence, the use of a definition file facilitates the introduction of new questions and the correction of errors. For example, after the software tool executes, it may become evident that there is an error, such as a misspelled word in a message provided to the student. The
- 25 misspelled word can be modified directly in the definition file, and the JAVA code run again, without having to recompile and/or rejar the software tool.

- Figure 9 illustrates the steps involved for an on-line educational course developer to create the list software tool of the illustrative embodiment of the present invention.
- 30 At some point in the educational course, the developer may include the list software tool of the present invention to test the user’s knowledge of the presented material. First, in step 80, the developer embeds a snippet in a course page of the on-line educational course for calling the list software tool. The snippet directs the browser and the user to

an HTML document containing the list software tool. For example, a suitable code fragment in a course page for calling a pop-up HTML page containing the software tool according to the teachings of the present inventions is as follows:

5

10

The above code allows a user to launch a pop-up page containing an examination question. The pop-up page of the illustrative embodiment has a width of 640 pixels and a height of 480 pixels.

15 Next, in step 81, the developer populates the question text and builds the pop-up page for the software tool. The pop-up page coding includes the above-described software tool tag or another suitable tag for executing a list software tool according to the teachings of the present invention. According to a preferred practice of the invention of the invention, the developer sets the width dimension of software tool in the software tool tag such that the function buttons (i.e. the check button and the reset button) are visible.

20 The developer creates a list definition file in step 82, including the desired parameters and tags, as described above. Finally, in step 83, the developer tests the functionality and desired presentation of the list software tool within an on-line educational course.

25 To implement the list software tool of the present invention, the list software tool includes computer executable instructions stored on a computer readable medium. The medium can include, for example, a hard disk, RAM medium, diskette, CD-ROM or other optical or magnetic storage medium. The instructions can be stored on a server that can be remote from the user. To run the application, the user can download the instructions to a computer readable medium of a local computer. The instructions can then be downloaded from the computer readable medium of the local computer to a local processor of the local computer, where the instructions are executed with the help of a virtual machine. A graphical user interface is generated by the instructions for

displaying a question and an answer box. The interface allows a user to select at least one answer selection to the question.

5 Although many of the foregoing examples have highlighted the use of JAVA in the present invention, other computer languages executed by virtual machines may be utilized for this purpose. As known to those of ordinary skill in the art, a virtual machine is an abstract computing machine having an instruction set and being capable of manipulating various memory areas at run time. A well-known virtual machine is the P-Code machine of UCSD Pascal, and the JAVA virtual machine. The JAVA virtual
10 machine does not assume any particular implementation technology, host hardware, or host operating system. It is not necessarily interpreted, and can be implemented by compiling its instruction set to that of a silicon processor. The JAVA virtual machine may also be implemented in microcode or directly in silicon.

15 These examples are meant to be illustrative and not limiting. The present invention has been described by way of example, and modifications and variations of the exemplary embodiments will suggest themselves to skilled artisans in this field without departing from the spirit of the invention. For example, the present invention is not limited to a JAVA software tool executed from a Web browser, and can include any
20 program executed from another application. The described software tool is not limited to execution from a pop-up Web page and can be embedded in any suitable document code in any suitable programming language. Alternately, the question and/or instructions to the user can be part of a definition file for the software tool, rather than a part of a static background created by a Web page or other suitable document.

25 Features and characteristics of the above-described embodiments may be used in combination. This description is to be construed as illustrative only and is for the purpose of teaching those skilled in the art the best mode for carrying out the invention. The preferred embodiments are merely illustrative and should not be considered
30 restrictive in any way. Details of the structure may vary substantially without departing from the spirit of the invention, and exclusive use of all modifications that come within the scope of the appended claims is reserved. It is intended that the invention be limited only to the extent required by the appended claims and the applicable rules of law. The

P5709
(SMQ-058)

scope of the invention is to be measured by the appended claims, rather than the preceding description, and all variations and equivalents that fall within the range of the claims are intended to be embraced therein.

C)
W)
CO
FI
LI
N)
PO
CI
S
W)
LI
LI
CI
FI
LI